ADVERSARIAL DEFECT SYNTHESIS FOR INDUSTRIAL PRODUCTS IN LOW DATA REGIME

Pasquale Coscia, Angelo Genovese, Fabio Scotti, Vincenzo Piuri

Department of Computer Science, Università degli Studi di Milano, Italy
{pasquale.coscia, angelo.genovese, fabio.scotti, vincenzo.piuri}@unimi.it

ABSTRACT

Synthetic defect generation is an important aid for advanced manufacturing and production processes. Industrial scenarios rely on automated image-based quality control methods to avoid time-consuming manual inspections and promptly identify products not complying with specific quality standards. However, these methods show poor performance in the case of ill-posed low-data training regimes, and the lack of defective samples, due to operational costs or privacy policies, strongly limits their large-scale applicability.

To overcome these limitations, we propose an innovative architecture based on an unpaired image-to-image (I2I) translation model to guide a transformation from a defect-free to a defective domain for common industrial products and propose simultaneously localizing their synthesized defects through a segmentation mask. As a performance evaluation, we measure image similarity and variability using standard metrics employed for generative models. Finally, we demonstrate that inspection networks, trained on synthesized samples, improve their accuracy in spotting real defective products.

***Index Terms*—** Synthetic defect generation, generative adversarial network, defective mask, residual network

1. INTRODUCTION

In today's competitive global market, industrial companies are increasingly challenged to deliver high-quality products and reliable services while complying with environmental policies [1]. Some operational methods, *e.g.*, green lean production [2], aim to reduce waste and pollution, optimizing resources and processes. Building automatic systems to be used on real production lines represents an important benefit in achieving these goals. Recent advancements in neural networks for classification [3] and recognition [4, 5] have found application in several academic and industrial fields. However, synthetic data generation in low data regimes still remains a challenging task [6]. More specifically, synthetic defect generation aims to create fake defects on products resembling real damages or inaccurate production steps. Regardless of its numerous applications, this topic suffers from attention due to limited publicly available resources (codes and datasets), and varieties of products and defects, making it difficult to define a standard benchmark for performance evaluation. Furthermore, designing a system able to automatically localize defects in synthetic samples could be extremely valuable in easily providing a visual cue on specific image regions. These systems may be a precious support for both humans and machines in pursuing advanced manufacturing.

Generative adversarial networks (GANs) [7] are currently employed for solving multiple tasks (*e.g.*, super-resolution [8], domain adaptation [9]) and demonstrated partial efficacy in generating reasonable and diverse defects for industrial products [10] for sufficiently large datasets, while approaches able to transform a source domain into a target domain appear to be more robust in low data regimes [11, 12, 10]. Previous methods partially address diversity and scalability issues related to the number of domains and styles considered. Choi *et al.* [13], for example, propose a single framework with multiple branches using the style information obtained from a latent code, or an input image, to generate multiple outputs. Rippel *et al.* [14] define a semantic-aware approach that employs a translation model to improve an anomaly detection classifier with pseudo-images representing synthetic defective samples. To control spatial and categorical characteristics as well, Zhang *et al.* [10] introduce a layer-wise composition into their encoder-decoder structure.

To handle products with different characteristics and address the scarcity of samples, we extend the CycleGAN architecture [12, 15], creating an unpaired image-to-image translation model. Our model transforms defect-free to defective domains and vice versa. In contrast to prior methods [10], we incorporate an additional branch to localize defective regions, extracting a segmentation mask that highlights defect locations. We utilize the MVTec Anomaly Detection (MVTec AD) dataset [16, 17], which offers high-quality images and pixel-wise annotated masks of various products, making it suitable for defect generation. Our approach accounts for

This work was supported in part by the EC under project EdgeAI (101097300), by the Italian MUR under PON project GLEAN, and by project SERICS (PE00000014) under the MUR NRRP funded by the EU - NextGenerationEU. Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the Italian MUR. Neither the European Union nor Italian MUR can be held responsible for them.

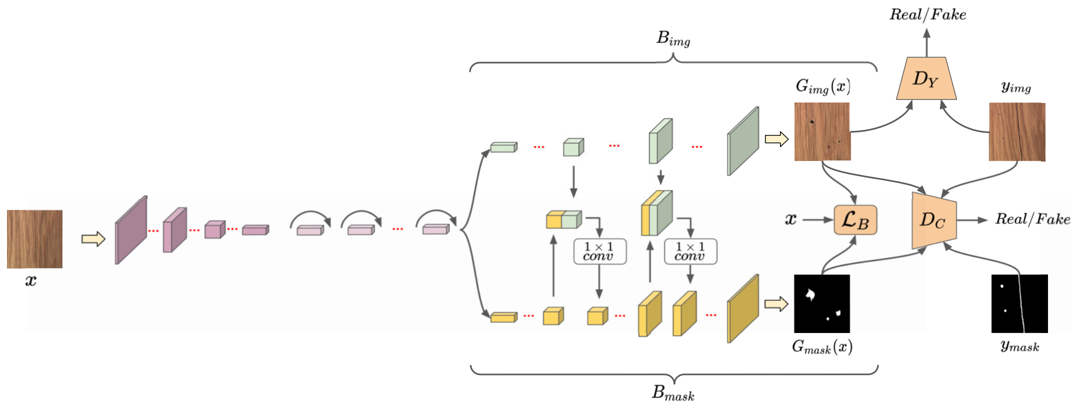

Fig. 1: Our residual-based generator consists of two connected up-sampling branches for reconstructing defective samples and corresponding segmentation masks. The inverse mapping F employs the same architecture.

diverse anomalies in shape, size, and structure, mirroring real-world industrial production scenarios. We validate our method by training deep defect inspection networks using our synthesized products, comparing it with other image generation approaches. The contribution of our work is two-fold. First, we build a novel residual-based convolutional neural network to generate defective samples and introduce losses capable of addressing imbalanced domains. Second, we simultaneously extract a segmentation mask that identifies defective regions. We demonstrate that our model is able to properly manage imbalanced domains and can be used to generate realistic defects.

The remainder of this paper is organized as follows. Section 2 presents our method. Section 3 discusses our results. Finally, Section 4 concludes the paper.

2. METHOD

Our goal is to map defect-free samples $x_i \in X$ to unpaired defective samples $y_i \in Y$. Since images of products without defects can be easily collected, our model can be used to obtain their corresponding defective counterparts. As proposed in Zhu *et al.* [12], we employ an adversarial loss to match the generated data distribution to the target domain and a cycle consistency loss to perform an unpaired translation. More in details, given two data distributions, $x \sim p_{norm}(x)$ and $y \sim p_{def}(y)$, two mappings are considered: $G : X \rightarrow Y$ and $F : Y \rightarrow X$. A discriminator D_X is used to distinguish between x and $F(y)$ while a discriminator D_Y between y and $G(x)$.

An adversarial loss discerns between real and synthesized defects as follows:

$$\mathcal{L}_{GAN} = \mathbb{E}_{x \sim p_{norm}(x)}[log(1 - D_Y(G(x))]+ \\ \mathbb{E}_{y \sim p_{def}(y)}[log D_Y(y)]. \quad (1)$$

A similar adversarial loss is used to discriminate between normal and synthesized defect-free samples. To restrict the space of possible mapping functions [12], forward and backward cycle consistency losses are employed:

$$\mathcal{L}_{cycle} = \mathbb{E}_{x \sim p_{norm}(x)}[||F(G(x)) - x||_1]+ \\ \mathbb{E}_{y \sim p_{def}(y)}[||G(F(y)) - y||_1]. \quad (2)$$

Fig. 1 depicts our residual-based convolutional neural network. Each block consists of a convolutional layer followed by instance normalization [18] and ReLU activation function. We use nine residual blocks [19] and two down-sampling and up-sampling blocks.

To reveal defective synthesized parts, we use an additional up-sampling branch, B_{mask}, coupled to the up-sampling image branch, B_{img}. More specifically, the output of each up-sampling B_{img} block is concatenated to its B_{mask} counterpart and followed by a 1x1 convolution to restore its previous features size. This embedding is then fed to the next up-sampling B_{mask} block. In this way, the mask is influenced by the corresponding defective synthesized image. To obtain masks that are coherent with the corresponding images, *i.e.*, masks reporting where a defect is actually localized, we introduce an additional consistency adversarial loss, based on a discriminator D_C, acting on a 4 channels map, to discriminate between real and synthesized concatenations of images and corresponding masks as follows:

$$\mathcal{L}_C = \mathbb{E}_{x \sim p_{norm}(x)}[log(1 - D_C(G_{img}(x)||G_{mask}(x))]+ \\ \mathbb{E}_{y \sim p_{def}(y)}[log D_C(y_{img}||y_{mask})], \quad (3)$$

where $G_{img}(x)$ and $G_{mask}(x)$ denote the synthesized image and mask, y_{img} and y_{mask} the real image and mask, and $||$ the concatenation operator. Since our primary aim is to investigate a mapping from a real to a defective domain, we employ the above loss only for the mapping G. Furthermore, to generate images with non-defective parts that closely resemble the source domain, we introduce an additional MSE loss $\mathcal{L}_B$. Specifically, given a defective generated mask $G_{mask}(x)$, we measure the distance between $o_b = x \odot (1 - G_{mask}(x))$ and $\hat{o}_b = G_{img}(x) \odot (1 - G_{mask}(x))$ where $\odot$ indicates a pixel-wise multiplication. This penalizes modifications to

Product	Defect	Training samples	Synthesized samples
Transistor	-	213	2
	Cut lead	8	60
Tile	-	230	3
	Cut	14	33
Wood	-	247	2
	Hole	8	19

Table 1: Number of training/synthesized samples per product and corresponding defects used for our analysis.

non-defective regions. Finally, we include an identity loss, as proposed in Zhu *et al.* [12].

The final objective function can be defined as follows:

$$\begin{aligned}
&\mathcal{L}(G, F, D_X, D_Y, D_C) = \\
&\quad \lambda_1 \mathcal{L}_{GAN}(G, D_Y, X, Y) \; + \; \lambda_2 \mathcal{L}_{GAN}(F, D_X, Y, X) \\
&\qquad + \; \lambda_{cycle} \mathcal{L}_{cycle}(G, F) \; + \; \lambda_{id} \mathcal{L}_{id}(G, F) \\
&\qquad\quad + \; \lambda_c \mathcal{L}_C(G, D_C, X, Y) \; + \; \lambda_b \mathcal{L}_B(o_b, \hat{o}_b), \qquad (4)
\end{aligned}$$

where different weights control the importance of each component. Therefore, we aim at solving:

$$\hat{G}, \hat{F} = \arg \min_{G,F} \max_{D_X, D_Y, D_C} \mathcal{L}(G, F, D_X, D_Y, D_C). \qquad (5)$$

3. EXPERIMENTAL RESULTS

To test our architecture with real industrial products, we rely on the MVTec AD dataset [16, 17]. Although intended for anomaly detection tasks, this dataset contains images that reflect defects from real production lines. The dataset includes 15 different objects with pixel-level annotations. For each product, a set of high-resolution defect-free training images and a test set containing both defective and defect-free images are provided. For the sake of simplicity, we limit our analysis to 3 products, as reported in Table 1.

Evaluation setting. To train our model, we use the training defect-free samples and 80% of defective samples. Our test set contains the testing defect-free images and the remaining defective samples. We limit our evaluation to the mapping G, *i.e.*, from a normal to a defective domain, as it is more important for real-world applications.

Evaluation metrics and baselines. Similarly to Sushko *et al.* [20], average LPIPS [21] and average LPIPS to the nearest image in the training set (Dist. to train) are evaluated. We also consider the FID [22] metric. We compare our model to Cycle-GAN [12], CUT [23] and its version without a regularizer FastCUT, NEGCUT [24], and a simple convolutional generative model, DCGAN [25], only trained on defective samples to synthesize images from a latent code. Cycle-GAN and DCGAN are modified to provide 4 channels output images (*i.e.*, images plus masks). For the other models, we use the original code as provided by the authors.

Training details. To train our model, we firstly downscale high-resolution input images and then randomly crop an area of 256×256 pixels. We also apply several data augmentation techniques (*e.g.*, flipping, normalization and elastic transformations) to both RGB images and masks. As discriminators, we use 70×70 PatchGAN [11] networks. As in Zhu *et al.* [12], the negative log likelihood objective is replaced by a least-squares loss. For our experiments, we set $\lambda_1 = \lambda_2 = 1$, $\lambda_{cycle} = 5$, $\lambda_{id} = 10$, $\lambda_c = 150$ and $\lambda_b = 0.05$ selected using a grid-search procedure. We use the Adam optimizer and a learning rate of 0.0002. This value is fixed for the first 100 epochs and linearly decays to zero over the next 200 epochs.

Results. We report our qualitative results in Fig. 2. Our model synthesizes realistic defects on different products more effectively. Both objects and textures are modified to introduce irregularities. Likewise, our segmentation masks highlight regions affected by defects, demonstrating the effectiveness in localizing these areas with a limited supervision. Compared to the baselines, our model also produces more high-quality images and less noisy segmentation masks. In this regard, CycleGAN appears unable to synthesize realistic defective masks, especially for textures, while DCGAN introduces noisy artifacts without properly affecting the input image. We note that our approach preserves the overall structure of the source domain. In Table 2 we provide a quantitative evaluation of these models. On average, our approach outperforms the baselines, especially for the FID metric, demonstrating higher fidelity in generating samples with similar characteristics to the training set. By contrast, low values of the LPIPS metric reported by the models indicate their limited capability in generating diverse samples.

We also conduct an ablation study to measure the impact of each introduced component in our architecture. As shown in Table 3, our complete architecture achieves the best FID metric. For each loss, we present a qualitative evaluation of their impact in Fig. 3.

Defect inspection analysis. Similarly to Zhang *et al.* [10], we also train two binary classifiers to demonstrate their effectiveness in spotting anomalies in real products by exploiting synthetic defective samples. Only real defect-free and synthetic defective images are used at training time. More specifically, we consider the following data split (train/val/test) for each experiment: $60/20/20$ for defect-free samples, $80/20/0$ for synthetic defective samples and $0/0/100$ for real defects. Table 4 confirms that synthetic samples significantly contribute to the identification of real defective samples.

4. CONCLUSION AND FUTURE WORK

Our approach synthesizes industrial defects with binary segmentation masks, mitigating the need for data collection. By extending an unpaired I2I translation framework with multi-branch generators and coherency loss, our model ensures compatibility between images and masks. It demonstrates superior effectiveness over conventional generative models, with improved metrics. Future work will explore applying it to scenarios with multiple defects and diverse domains.

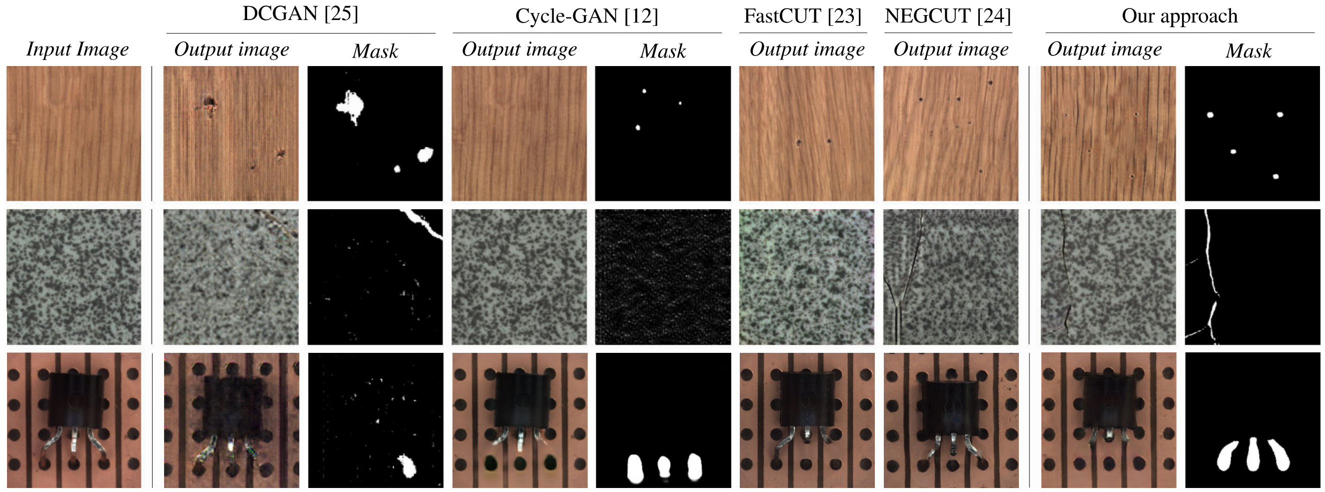

Fig. 2: Qualitative results for different input images of some baselines and our model. Our method provides better results and more robust segmentation masks.

Method	Product (Defect)	FID ↓	LPIPS ↑	Dist. to train
DCGAN [25]	Transistor (Cut lead)	7.39	0.03	0.18
	Tile (Crack)	5.82	0.02	0.16
	Wood (Hole)	4.56	0.00	0.16
	Avg	5.92	0.02	0.17
CycleGAN [12]	Transistor (Cut lead)	0.22	0.10	0.11
	Tile (Crack)	1.80	0.20	0.23
	Wood (Hole)	5.78	0.11	0.18
	Avg	2.60	0.14	0.17
CUT [23]	Transistor (Cut lead)	0.54	0.13	0.09
	Tile (Crack)	2.55	0.14	0.26
	Wood (Hole)	8.24	0.06	0.21
	Avg	3.78	0.11	0.19
FastCUT [23]	Transistor (Cut lead)	1.36	0.17	0.13
	Tile (Crack)	5.06	0.20	0.24
	Wood (Hole)	10.37	0.19	0.19
	Avg	5.60	**0.19**	0.19
NEGCUT [24]	Transistor (Cut lead)	0.80	0.13	0.10
	Tile (Crack)	2.81	0.26	0.28
	Wood (Hole)	6.90	0.06	0.20
	Avg	3.50	0.15	0.19
Proposed approach	Transistor (Cut lead)	0.17	0.12	0.11
	Tile (Crack)	1.75	0.20	0.24
	Wood (Hole)	2.25	0.23	0.22
	Avg	**1.39**	0.18	0.19

Table 2: Quantitative results. Our method reports more fidelity in synthesizing industrial defects compared to the baselines.

	Transistor (Cut lead)					
Model	$\mathcal{L}_{GAN}+\mathcal{L}_{cycle}+\mathcal{L}_{id}$	$\mathcal{L}_C$	$\mathcal{L}_B$	FID ↓	LPIPS ↑	Dist. to train
(a)	✓	✗	✗	0.80	0.12	0.10
(b)	✓	✓	✗	0.73	0.11	0.12
(c)	✓	✗	✓	0.23	**0.13**	0.10
(d)	✓	✓	✓	**0.17**	0.12	0.11

Table 3: Ablation study for our losses. Our complete model achieves the best performance.

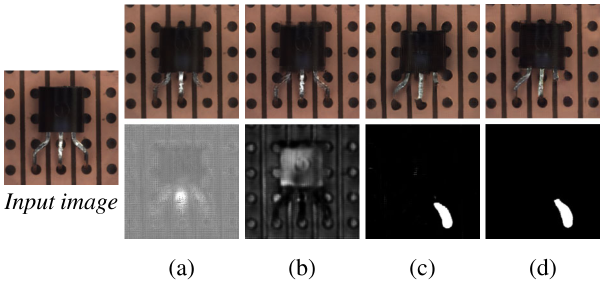

Fig. 3: Qualitative results for the ablation study reported in Table 3 where each loss is added to our model. For visualization purposes, we show the output of the sigmoid layer of the mask branch.

Backbone	Product (Defect)	Accuracy (%)	Precision	Recall
-	Transistor (cut lead)	84.61	0.42	0.50
	Wood (hole)	84.37	0.42	0.50
	Tile (crack)	75.71	0.38	0.50
	Avg	81.56	0.41	0.50
ResNet-18 [19]	Transistor (cut lead)	90.77	0.95	0.70
	Wood (hole)	93.75	0.88	0.88
	Tile (crack)	87.50	0.94	0.60
	Avg	88.60	0.88	0.66
DenseNet-121 [26]	Transistor (cut lead)	95.38	0.97	0.85
	Wood (hole)	90.62	0.95	0.70
	Tile (crack)	89.06	0.94	0.65
	Avg	**91.69**	**0.95**	**0.73**

Table 4: Defect inspection results. The first group denotes a classifier that outputs a defect-free class regardless of its input. Both networks increase their accuracy using our synthesized samples.

5. REFERENCES

[1] European Commission and Executive Agency for Small and Medium-sized Enterprises, A Doranova, M Mueller, R Zhechkov, K Izsak, and L Roman, *Green action plan for SMEs: addressing resource efficiency challenges and opportunities in Europe for SMEs : implementation report*, Publications Office, 2018.

[2] Stefano Saetta and Valentina Caldarelli, "Lean production as a tool for green production: the green foundry case study," *Procedia Manufacturing*, 2020.

[3] Mazda Moayeri, Phillip Pope, Yogesh Balaji, and Soheil Feizi, "A comprehensive study of image classification model sensitivity to foregrounds, backgrounds, and visual attributes," in *CVPR*, 2022.

[4] Junho Kim, Jaehyeok Bae, Gangin Park, Dongsu Zhang, and Young Min Kim, "N-imagenet: Towards robust, fine-grained object recognition with event cameras," in *ICCV*, 2021.

[5] Adrian Ziegler and Yuki M. Asano, "Self-supervised learning of object parts for semantic segmentation," in *CVPR*, 2022.

[6] Shancong Mou, Meng Cao, Zhendong Hong, Ping Huang, Jiulong Shan, and Jianjun Shi, "Synthetic Defect Generation for Display Front-of-Screen Quality Inspection: A Survey," *arXiv e-prints*, 2022.

[7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *NeurIPS*, 2014.

[8] Yiqun Mei, Yuchen Fan, and Yuqian Zhou, "Image super-resolution with non-local sparse attention," in *CVPR*, 2021.

[9] Shuang Li, Mixue Xie, Fangrui Lv, Chi Harold Liu, Jian Liang, Chen Qin, and Wei Li, "Semantic concentration for domain adaptation," in *ICCV*, 2021.

[10] Gongjie Zhang, Kaiwen Cui, Tzu-Yi Hung, and Shijian Lu, "Defect-gan: High-fidelity defect synthesis for automated defect inspection," in *WACV*, 2021.

[11] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros, "Image-to-image translation with conditional adversarial networks," in *CVPR*, 2017.

[12] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *ICCV*, 2017.

[13] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha, "Stargan v2: Diverse image synthesis for multiple domains," in *CVPR*, 2020.

[14] Oliver Rippel, Maximilian Müller, and Dorit Merhof, "Gan-based defect synthesis for anomaly detection in fabrics," in *ETFA*, 2020.

[15] Angelo Genovese, Vincenzo Piuri, and Fabio Scotti, "Towards explainable face aging with generative adversarial networks," in *ICIP*, 2019.

[16] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger, "Mvtec ad – a comprehensive real-world dataset for unsupervised anomaly detection," in *CVPR*, 2019.

[17] Paul Bergmann, Kilian Batzner, Michael Fauser, David Sattlegger, and Carsten Steger, "The mvtec anomaly detection dataset: A comprehensive real-world dataset for unsupervised anomaly detection," *IJCV*, 2021.

[18] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky, "Instance Normalization: The Missing Ingredient for Fast Stylization," *arXiv e-prints*, 2016.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[20] Vadim Sushko, Jurgen Gall, and Anna Khoreva, "One-shot gan: Learning to generate samples from single images and videos," in *CVPRW*, 2021.

[21] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, 2018.

[22] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *NeurIPS*, 2017.

[23] Taesung Park, Alexei A. Efros, Richard Zhang, and Jun-Yan Zhu, "Contrastive learning for unpaired image-to-image translation," in *ECCV*, 2020.

[24] Weilun Wang, Wengang Zhou, Jianmin Bao, Dong Chen, and Houqiang Li, "Instance-wise hard negative example generation for contrastive learning in unpaired image-to-image translation," in *ICCV*, 2021.

[25] Alec Radford, Luke Metz, and Soumith Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *ICLR*, 2016.

[26] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger, "Densely connected convolutional networks," in *CVPR*, 2017.